# IMPLEMENTASI LAYANAN PESAN PENDEK SEBAGAI PIRANTI NOTIFIKASI SISTEM DETEKSI INTRUSI

**Wahyu Sapto Aji[1], Soedjatmiko[2], Sujoko Sumaryono[3]**
[1]Program Studi Teknik Elektro, Universitas Ahmad Dahlan, Jln. Prof. Soepomo, Yogyakarta
[2,3]Program Studi Teknik Elektro, Universitas Gadjah Mada, Jln. Grafika, Yogyakarta
e-mail: , wahyusaji@yahoo.com[1], djatmiko@te.ugm.ac.id[2], sujoko@te.ugm.ac.id[3]

***Abstrak***

*Penelitian ini mengkaji penerapan layanan pesan pendek (SMS) sebagai piranti notifikasi sistem deteksi intrusi/penyusupan (IDS) dalam usaha untuk meningkatkan keterawasan sistem. Penelitian dilakukan dengan membangun piranti SMS Gateway yang akan memberikan informasi berupa SMS jika terjadi kegiatan yang dianggap sebagai usaha percobaan intrusi. Sistem notifikasi ini juga diharapkan memberikan kemampuan administrator melakukan tindakan terbatas terhadap server apabila terjadi kegiatan intrusi tadi. Hasil penelitian menunjukkan telah dapat dikembangkan piranti notifikasi SMS. Waktu rerata yang dibutuhkan untuk menyampaikan informasi kepada administrator dalam simulasi serangan yang dilakukan berkisar pada angka 1.6 menit. Dalam pengujian dilakukan simulasi tindakan berupa pematian server, penguncian server, penyambungan dan pemutusan server ke internet dan reboot server.*

***Kata kunci:*** *IDS, SMS, keamanan jaringan, administrator, internet, intrusi, notifikasi, remote*

***Abstract***

*This research investigates the implementation of SMS (Short Message Service) as notification tool of an IDS (Intrusion Detection System) in term to improve network reliability. The research has doing by build a SMS Gateway . The purpose of SMS Gateway is to send network administrator if there is exist an activity that being considered as a part of intrusion activity. SMS Gateway should be give the network administrator capability to take action over network server via SMS remotely. The result of research shows that such of SMS gateway has been built successly. Average times that SMS Gateway needed to send its information is at 1.6 minutes. In system testing, the network administrator capable to take action over the server via SMS remotely. The action that simulated is turn off server, lock server, connect and disconnect to the internet and reboot the server.*

***Keywords:*** *IDS, SMS, network security, administrator, internet, intrusion, notification, remote*

## 1. PENDAHULUAN

Sejak kemunculannya dalam *Internet World Expo* ditahun 1996, internet mendapat sambutan luas dan mengalami pertumbuhan yang sangat pesat dari tahun ke tahun. Kekuatannya dalam menyampaikan informasi dan kemampuannya menjebatani komunikasi teks, suara bahkan visual, menjadikan internet sebagai salah satu media informasi dan komunikasi yang sangat penting sekarang ini.

Akan tetapi dengan kondisi internet yang terbuka, berarti menyimpan kemungkinan ancaman atau bahaya bagi sistem yang terhubung ke internet. Hal tersebut sangat disadari oleh pengguna internet dan banyak usaha dilakukan untuk mencegah ancaman tadi menjadi kenyataan.

Sebuah kajian tentang keamanan jaringan yang dilakukan Yegeneswaran[1] memperkirakan usaha serangan instrusi di internet ditahun 2003 adalah berkisar pada angka 25 milyar usaha serangan perharinya dan cenderung terus meningkat.

Menurut catatan CERT/CC [2], menunjukan hal serupa dimana insiden serangan atau ancaman yang dilaporkan dari tahun 1988 sampai tahun 2002 naik dengan cepat dan cenderung membentuk kurva eksponesial.

## 2. METODE PENELITIAN

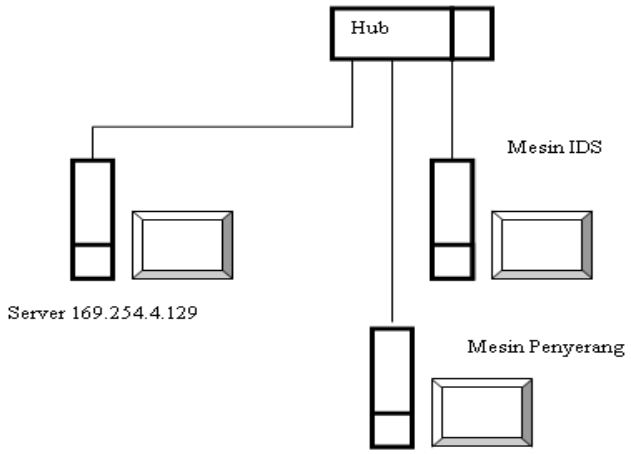
### 2.1. Instrumen Penelitian

Untuk mewujudkan rancangan sistem integrasi notofikasi SMS dalam IDS, diperlukan alat bantu utama dan alat-alat penguji antara lain :

a. PC (*Personal Computer*) Pentium IV dengan frekuensi detak sebesar 2100 MHz, Kapasitas RAM 240 MB. Alat bantu ini digunakan untuk menulis program dalam bahasa Delphi.
b. Bahasa pemrograman Delphi, bahasa pemrograman ini digunakan untuk membuat program dalam penelitian. Versi yang digunakan dalam penelitian adalah Borland Delphi Versi 7 ditambah dengan piranti VCL dari Scibit untuk menghubungkan database MySQL dengan program.
c. Piranti lunak WinPcap. Piranti lunak Winpcap digunakan untuk menangkap data dalam jaringan.
d. Piranti lunak WinSnort. Winsnort merupakan piranti lunak IDS .
e. Piranti lunak IIS 5.1. Piranti IIS 5.1 atau *Internet Information Service* 5.1 digunakan sebagai web server.
f. Mysql. Piranti lunak Mysql digunakan sebagai pengelola database yang dibangkitkan oleh WinSNort.
g. Interbase. Interbase digunakan untuk pengelolaan database SMS manager.
h. PHP. Piranti lunak PHP digunakan untuk pembuatan script web.
i. TOxygenSMS. ToxygenSMS merupakan komponen VCL yang menjebatani komunikasi antara pesawat telepon genggam dengan komputer. Pemilihan ToxyenSMS disebabkan karena kestabilan yang sudah teruji.

### 2.2. Pelaksanaan Penelitian

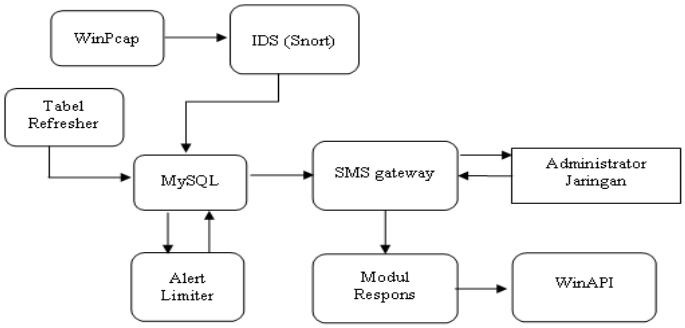
Skema jaringan yang dipergunakan pada penelitian in adalah ditunjukkan pada Gambar 1.




Gambar 1. Skema Jaringan dalam Penelitian

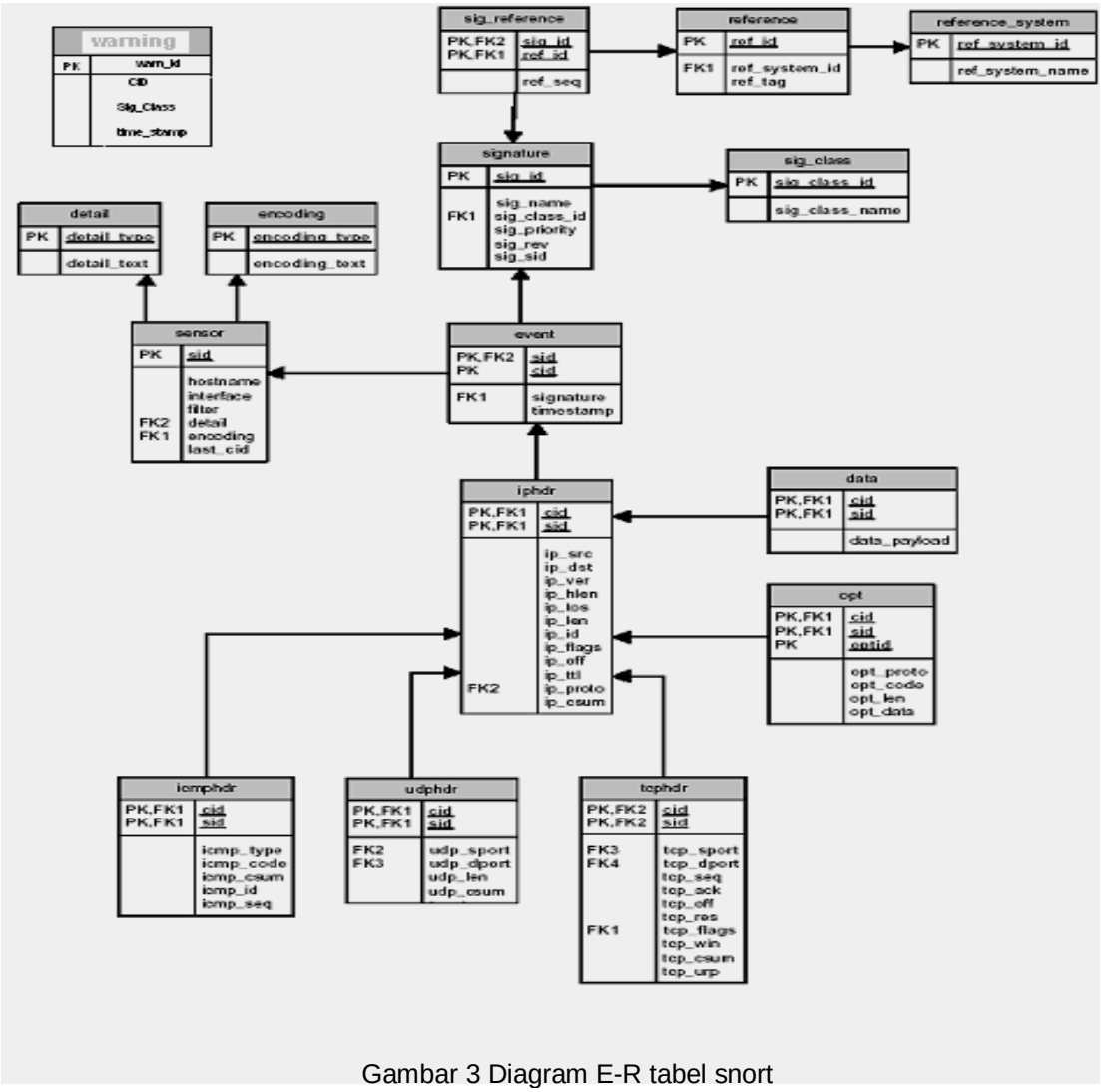
Jaringan yang digunakan dalam percobaan merupakan jaringan lokal intranet. Media konektor jaringan menggunakan kabel UTP dan dengan kartu Ethernet sebagai kartu adapter jaringan. Dalam skema diatas, Snort yang diinstal difungsikan sebagai pemonitor lalu lintas data menuju server.

Konfigurasi perangkat lunak pada penelitian ini melibatkan beberapa software. Gambar 2 menampilkan konfigurasi software yang digunakan pada penelitian ini dalam diagram blok.




Gambar 2. Diagram blok sistem

Garis besar kerja dari sistem Gambar 2 adalah bahwa data yang ditangkap oleh WinPcap akan diolah oleh Snort sesuai dengan *rule* yang telah ditentukan. Alert yang dibangkitkan oleh Snort akan disimpan didalam database dengan menggunakan server database MySQL. Oleh sub-program *alert limiter*, database alert yang tersimpan dalam snort difilter, hasil filter ini ditampung ke dalam tabel yang diberi nama tabel warning, sehingga diharapkan tidak terjadi *sms flooding* ke administrator.




Gambar 3 Diagram E-R tabel snort

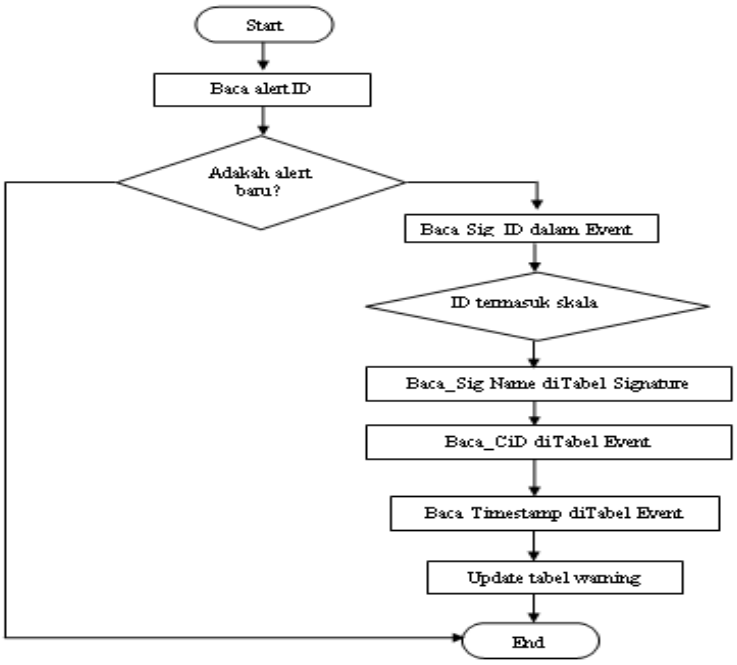
Setelah terjadi pembaharuan tabel warning, maka SMS gateway akan membaca tabel warning kemudian mengirimkannya ke administrator. Karena kemungkinan adanya pembengkakkan memori yang diperlukan untuk menyimpan isi tabel-tabel ini, maka isi tabel di atas akan dihapus setelah 24 jam. Tindakan ini perlu dilakukan mengingat kapasitas memori pada mesin yang terbatas yaitu sekitar 6 Gbyte. Sub-program ini diberi nama Tabel Refresher.

Modul respons merupakan modul program yang berfungsi melakukan aksi mematikan server, mereboot server, memutus hubungan ke internet, menyambung hubungan ke internet, dan mengunci server. Aksi yang dijalankan dilakukan berdasarkan intruksi dari pesawat telepon genggam administrator system.

**a. Pembentukan dan Konfigurasi Tabel**

Pada penelitian ini server database yang dipilih adalah MySQL, Untuk koneksi database server dengan program digunakan driver MyODBC. Setelah terkoneksi dengan driver MyODBC koneksi selanjutnya menggunakan komponen ADO yang sudah tersedia dalam *pallete* Delphi . Tabel yang dibentuk dalam MySQL diletakkan database bernama Snort.

Konfigurasi tabel pada penelitian ini dapat digambarkan dalam diagram E-R (Entity Relationship) Gambar 3.




Gambar 4. Flowchart pembaharuan isi tabel warning

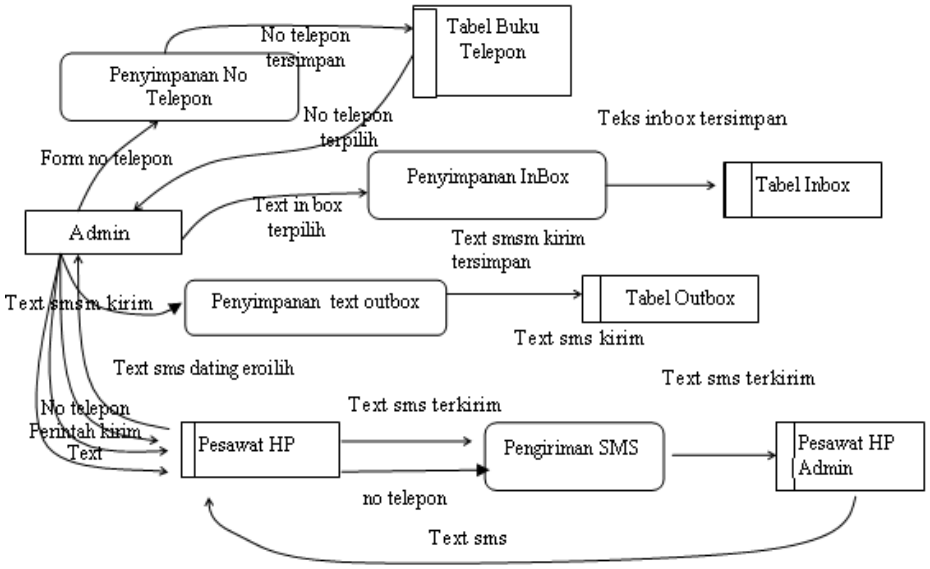
**b. Alert Limiter**

Usaha untuk membatasi alert yang dibangkitkan oleh snort sudah dimulai dari file konfigurasi snort. File konfigurasi snort, pada bagian unit diatur untuk membangkitkan alert dengan mengikuti ambang tertentu. Selain memanfaatkan unit *treshold* dalam Snort, untuk membatasi pesan serangan yang dikirim, dilakuan pembentukan tabel warning. Tabel warning ini berisikan alert serangan dengan prioritas maksimal. Flow chart untuk *updating* dari isi tabel warning ditunjukkan pada Gambar 4.

Guna mendapatkan nama signature dari serangan terakhir dilakukan operasi query terhadap tabel signature dengan parameter bahwa sig_id dari tabel signature adalah sama dengan signature saat serangan terakhir atau saat cid dari tabel event memilki harga maksimal. Untuk mengurangi jumlah alert yang akan dikirimkan ke SMS maka sebelum pengiriman dilakukan dengan mengecek skala prioritas dari serangan. Dalam penelitian ini SMS akan dikrimkan apabila skala prioritas serangan adalah memiliki skala 1 atau 2. Untuk mendapatkan posisi terakhir dari record dipakai fungsi *recordcount*, atau pakai fungsi *last* lalu diuji dengan *eof*.

**c. SMS Manager**

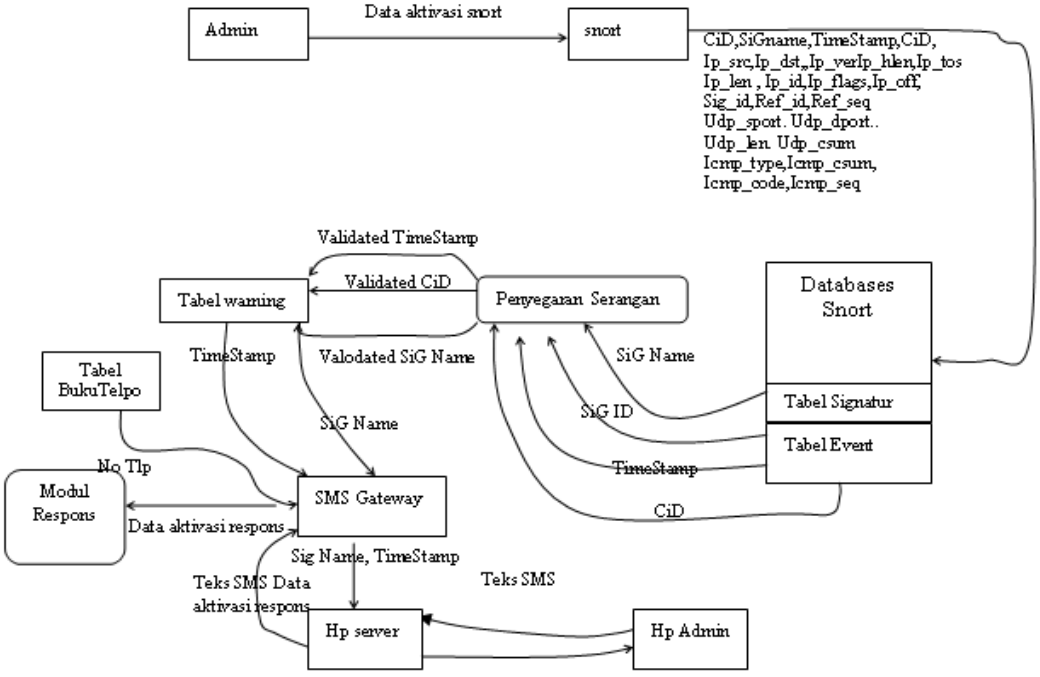
SMS Manager disini berfungsi mengatur fitur –fitur yang terdapat dalam program yaitu pengaturan buku telepon, pengaturan pesan masuk dan pengaturan pesan keluar. Diagram DFD dalam SMS Manager ini dapat dilihat dalam Gambar 5.




Gambar 5 Diagram DFD SMS Manager

**d. IDS Manager**

Bagian IDS manager merupakan bagian yang berfungsi untuk melakukan aktivasi IDS. DFD dari bagian IDS manager ditunjukkan pada Gambar 6.




Gambar 6. Diagram DFD IDS Manager

Pada Gambar 6 tersebut, seorang admin melakukan aktivasi sistem serta melakukan pengeditan terhadap tabel buku telepon. Setelah aktivasi sistem, maka akan terjadi pembaruan tabel warning dalam selang waktu yang telah ditentukan. Apabila terjadi penambahahan tabel warning dan setelah dilakukan pengujian terhadap SiG dan ternyata merupakan serangan baru, maka dilakukan pengiriman SMS ke administrator.

Modul respons berfungsi untuk melakukan aksi tertentu. Aksi ini dikirimkan oleh administrator melewati pesawat telepon genggam admin.

**e. Pembentukan server victim**

Jaringan yang dibentuk dalam percobaan ini merupakan jaringan intranet local dengan sebuah server korban dan 2 buah server penyerang. Server korban ini memiliki alamat http:\\ victim\ dengan alamat IP 169.254.11.93. Sistem operasi dari server korban adalah Windows XP SP1 dan menjalankan IIS 5.0 sebagai web server. Selain berfungsi sebagai web server, server korban juga menyediakan layanan SQL server dengan menjalankan program MySQL. Dalam penelitian ini MySQL menjalankan database Situsku yang didalamnya terdapat sebuah tabel bernama tabel Anggota. Tujuan pembentukan tabel anggota adalah untuk melakukan simulasi serangan *SQL attack*. Struktur tabel Anggota memiliki 4 kolom yaitu kolom nomer, nama, alamat dan Nomor Telepon. Dalam web server korban juga disediakan halaman PHP yang bertugas untuk melakukan query ke database Situsku.

**f. Pembentukan host penyerang**

Host penyerang berfungsi untuk menjalankan serangan ke server korban. Dalam penelitan ini digunkan dua buah komputer penyerang yang masing-masing bernama http:\\laptop\ dengan alamat ip 169.254.44.92 danhttp:\\teuad2\ dengan alamat ip 169.254.11.94

**g. Konfigurasi Snort**

Pada penelitian ini konfigurasi preposesor yang digunakan adalah sebagai berikut:

- Preprossesor Frag3. Konfigurasi preprosesor frag3 bertujuan untuk defragmentasi IPoleh Snort.
- Preprocessor Stream4. Preprossesor ini digunakan untuk penyusunan ulang aliran paket data TCP.
- Preprossesor sfPortscan. Preprossesor ini digunakan untuk mendeteksi usaha yang berupa s*canning* port dari korban.

Snort dikonfigurasi agar mendukung operasi plug-in ke dalam database MySQL. Sintak untuk konfiguasri plug-in adalah sebagai berikut:

```
output database: log, mysql, user=snort password=logger
        dbname=snort host=localhost port=3306
```

Tipe output adalah log, hal ini berarti informasi yang didapat Snort akan dimasukkan ke dalam tabel.. Database yang digunakan adalah database yang diberi nama Snort. Selain menggunakan database, output plug in dari snort juga dikonfigurasi untuk menggunakan file log, sdengan sintak konfigurasi sebagai berikut:

```
output ALERT_FAST: alert.ids
```

Alert_fast menunjukkan tipe informasi yang dipilih, sedangkan alert.ids merupakan nama file log tempat menyimpan data. Posisi file log ditentukan dalam perintah saat menjalankan Snort. Untuk menjalankan Snort dalam mode IDS dijalankan perintah:

```
Snort -i2 -c D:\win-ids\snort\etc\snort.conf -l D:\win-ids\snort\mysql\data\snort
```

**h. SMS Gateway**

Komponen SMS gateway ini berfungsi sebagai pengirim pesan peringatan ke handphone Admin.Tabel yang digunakan dalam komponen SMS gateway meliputi:

- Tabel phone book

– Tabel warning

Kedua tabel tersebut diatas tidak memiliki hubungan E-R sehingga masing-masing berdiri sendiri.

## 3. HASIL DAN PEMBAHASAN

Pengujian sistem dilakukan dengan cara melakukan simulasi serangan atau aktifitas yang nantinya akan dideteksi oleh sensor dan sistem menanggapinya dengan mengirim pesan ke administrator.

### 3.1. Menjalankan Snort

Dalam percobaan interface jaringan yang digunakan adalah berupa Ethernet. Snort dijalankan dengan mode IDS dan mode output logging ke database Mysql yang diberi nama Snort. Untuk menjalankan mode tersebut sintak command prompt dari adalah sebagai berikut:

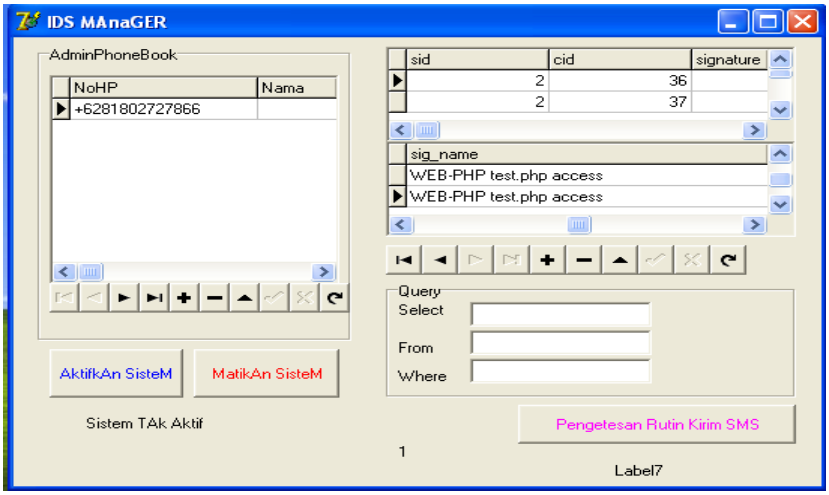
```
Snort -i2 -iv -c D:/win-ids/snort/etc/snort.conf -l D:/win-ids/snort/log/alert.
```

### 3.2. Probing PHP Test

Aktifitas probing dengan tujuan untuk mendapatkan informasi tertentu dari web korban biasanya merupakan awal dari kegiatan untuk melakukan kompromisasi web korban. Dalam simulasi ini dibuat dengan skenario pemberian perintah test.php untuk memanfaatkan fungsi phpnfo() oleh penyerang terhadap web server korban. Guna mendeteksi serangan diatas aturan snort yang digunakan dalam percobaan adalah sebagai berikut:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-PHP test.php access";
      flow:to_server,established; uricontent:"/test.php"; reference:nessus,11617;
                classtype:web-application-activity; sid:2152; rev:1;)
```

Serangan ini dapat ditangkap oleh IDS manager. Gambar 7 menampilkan tampilan IDS manager saat menangkap usaha probing dalam simulasi ini.




Gambar 7. Tampilan IDS Manager saat menangkap usaha probing

Pada Gambar 7, nomor pesawat telepon genggam dalam simulasi adalah 081802727866. Serangan ini tertangkap dengan nama serangan WEB PHP test_php accses dengan waktu saat terjadinya serangan adalah tanggal 14/o6/2007 pada pukul 09:49:00 AM.

IDS manager berhasil mengirimkan informasi adanya upaya probing ke pesawat telepon genggam administrator, hal ini ditunjukkan dalam Gambar 7. Pada Gambar tersebut, pesan diterima oleh admin pada tanggal 14/06/2007 dan pada pukul 09:51: 16 dengan signature serangan test php acces.


a b

Gambar 7. Tampilan pesan di pesawat Admin

### 3.3. Serangan SQL Injection

Simulasi serangan SQL injection dilakukan dengan cara mesin penyerang memberikan perintah URL ke server korban dengan sintak sebagai berikut:

```
http:\\victim\nama.php? nama=joni;or 1=1
http:\\victim\nama.php? nama=joni;  ”or 1=1
http:\\victim\nama.php? nama=joni;  ”1=1–
http:\\victim\nama.php? nama=joni;  ‘or’a=’a
http:\\victim\nama.php? nama=joni;  ”or”a=”a
http:\\victim\nama.php? nama=joni;  ”or 0=0--
http:\\victim\nama.php? nama=joni;  or 0=0
http:\\victim\nama.php? nama=joni;  ”or 0=0#
http:\\victim\nama.php? nama=joni;  or 0=0 #
http:\\victim\nama.php? nama=joni;  or 0=0 #
```

Pada serangan tersebut server htpp:\victim\ memiliki tabel anggota dengan salah satu nama kolomnya adalah ‘nama’.

Sidik untuk mendeteksi serangan ini adalah dengan cara mengenali karakter khusus yang disebut sebagai *SQL meta-charackter*. Karakter ini tergantung dari server database yang digunakan. Sidik dari serangan SQL ini yang dicobakan dalam penelitian ini adalah sebagai berikut:

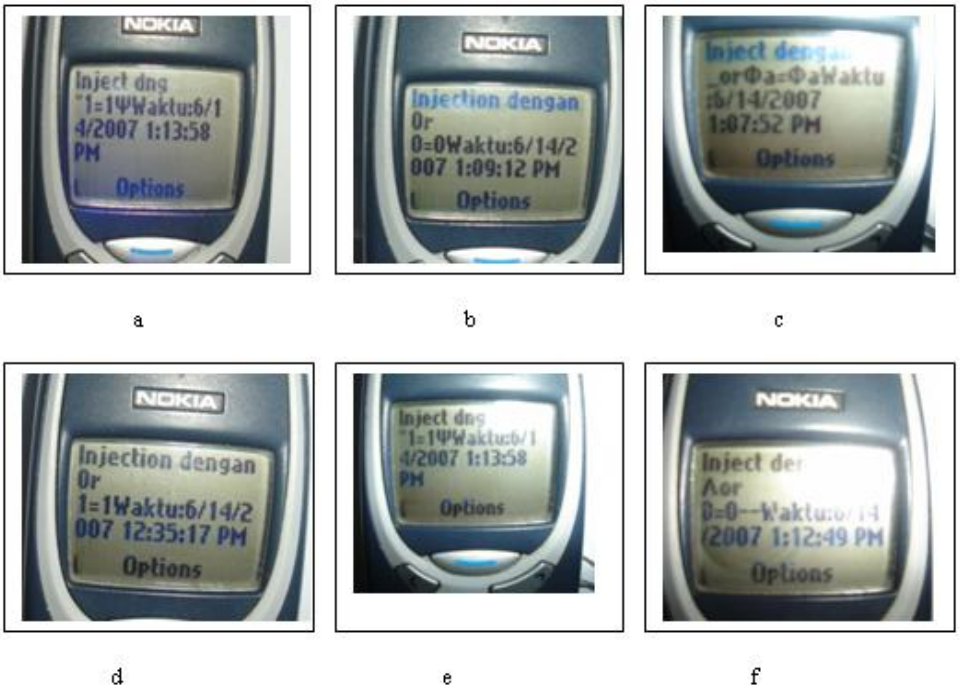
a. **/((\%3D)|(=))[^\n]*((\%27)|(\')|(\-\-)|(\%3B)|(;))/i**
   Sidik ini pertamakali mencari adanya karakter = baik dalam bentuk aslinya atau dalam bentuk ekuivalen hexadecimal (%3D) kemudian mengecek adanya karakter *single quote* , *double dash* ataupun karakter semi-colon. Sidik ini untuk mengenali serangan SQL yang memanfaatkan karakter *single quote* untuk memanipulasi query asli sehingga selalu menghasilkan nilai kebenaran yang selalu benar.
b. **/\w*((\%27)|(\'))((\%6F)|o|(\%4F))((\%72)|r|(\%52))/ix**
   Sidik serangan ini merupakan perbaikan dari sidik sebelumnya , yaitu dengan menambahkan pengecekan kata ‘or’ baik dalam versi aslinya maupun dalam versi hexadecimal, yaitu dengan memasukkan statemen (\%6F)|o|(\%4F))((\%72)|r|(\%52).
c. ***/((\%27)|(\'))union/ix(\%27)|(\')***
   Sidik ini digunakan untuk mendeteksi usaha serangan SQL dengan menggunakan perintah query ‘union’
d. ***/exec(\s|\+)+(s|x)p\w+/ix***
   Sidik ini digunakan untuk mendeteksi usaha serangan SQL terhdap sistem yang menggunakan MSSQL sebagai server database serta Windows XP sebagai sistem operasinya dan penyerang berusaha untuk melakukan eksekusi prosedur yang berbahaya.

Guna mendeteksi ancaman di atas disusun aturan dalam Snort sebagai berikut:

```
#-----------------------------------
# RULE SIMULASI SERANGAN SQL INJECTION
#-----------------------------------
```

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"SQL Injection -
Paranoid"; flow:to_server,established;uricontent:".php";pcre:"/(\%27)|(\')|(\-\-
)|(%23)|(#)/i"; classtype:Web-application-attack; sid:9099; rev:5;)
alert tcp 169.254.223.99 any -> $HTTP_SERVERS $HTTP_PORTS (msg:"PERCOBAAN SQL
INJECTION DENGAN OR 1=1"; flow:to_server,established; uricontent:"or 1=1"; nocase;
classtype:web-application-activity; sid:1)
alert tcp 169.254.223.99 any -> $HTTP_SERVERS $HTTP_PORTS (msg:"PERCOBAAN SQL
INJECTION DENGAN OR 1=1--"; flow:to_server,established; uricontent:"or 1=1--";
nocase; classtype:web-application-activity; sid:2)
alert tcp 169.254.223.99 any -> $HTTP_SERVERS $HTTP_PORTS (msg:"PERCOBAAN SQL
INJECTION DENGAN "OR 1=1"; flow:to_server,established; uricontent:" "or 1=1";
nocase; classtype:web-application-activity; sid:3)
alert tcp 169.254.223.99 any -> $HTTP_SERVERS $HTTP_PORTS (msg:"PERCOBAAN SQL
INJECTION DENGAN "OR 0=0--"; flow:to_server,established; uricontent:"or 1=1";
nocase; classtype:web-application-activity; sid:1)
alert tcp 169.254.223.99 any -> $HTTP_SERVERS $HTTP_PORTS (msg:"PERCOBAAN SQL
INJECTION DENGAN OR 0=0#"; flow:to_server,established; uricontent:"or 1=1";
nocase; classtype:web-application-activity; sid:1)
```

Pada aturan di atas, alamat IP 169.254.223.99 merupakan alamat penyerang sedangkan alamat IP 169.254.11.93 merupakan alamat IP korban , yang dalam hal ini memiliki URL http:\victim. Kata kunci yang digunakan dalam mendeteksi ini adalah kata kunci uricontent, kata kunci ini bertugas mengenali penggunaan sintak query yang tidak umum seperti di atas. IDS manager berhasil menangkap serangan ini dan mengirimkan informasinya ke dalam pesawat telepon genmggam admin. Gambar 8 menunjukkan tampilan pesawat telepon genggam setelah menerima pesan adanya usaha untuk melakukan serangan SQL Injection.




Gambar 8. Tampilan pesan pada HP saat simulasi serangan SQL injection

Hasil pengujian serangan terhadap sistem menghasilkan data yang ditampilkan dalam Tabel 1.

Tabel 1. Hasil uji sistem terhadap simulasi serangan *SQL Injection*

| Nama Serangan | Jam Serangan | Jam Kirim SMS | *Timeliness* | Operator Seluler |
|---|---|---|---|---|
| SQl attack 1 | 12.01 | 12.02 | 1 menit | Pro XI |
| SQl attack 2 | 12.07 | 12.08 | 1 menit | Pro XI |
| SQl attack 3 | 12.11 | 12.12 | 1 menit | Pro XI |
| SQl attack 4 | 12.19 | 12.21 | 2 menit | Pro XI |
| SQl attack 4 | 12.27 | 12.28 | 1 menit | Pro XI |
| SQl attack 5 | 12.35 | 12.36 | 1 menit | Pro XI |
| SQl attack 6 | 12.45 | 12.47 | 2 menit | Pro XI |
| SQl attack 7 | 12.57 | 12.58 | 1 menit | Pro XI |
| SQl attack 8 | 13.05 | 13.07 | 2 menit | Pro XI |
| SQl attack 9 | 13.18 | 13.20 | 2 menit | Pro XI |
| SQl attack 10 | 13.32 | 13.33 | 1 menit | Pro XI |
| SQl attack 11 | 13.50 | 13.51 | 1 menit | Pro XI |

Pada hasil pengujian ini, dapat ditunjukkan bahwa sistem telah berhasil dalam mengirimkan pesan peringatan ke administrator saat sensor IDS mendeteksi adanya serangan. Rentang waktu yang terbentang antara saat terjadinya serangan dengan pengiriman SMS tentunya merupakan fungsi dari lalu lintas data operator seluler yang bersangkutan.

### 3.4. Serangan *Cross Site Scripting* (XSS)

Fase awal serangan *Cross Site Scripting* umumnya adalah usaha penyerang untuk melakukan pengecekkan sebuah website memiliki lubang terhadap serangan XSS. Usaha test ini untuk mendapatkan pesan tertentu saat webserver mendapat karakter serangan. Teks untuk probing ini dapat berupa script yang bersifat coba-coba seperti <script>alert('document.cookie')</script> atau menggunakan tag HTML seperti <b>, atau <u>. Bagi penyerang, umumnya untuk menghindari usaha pelacakan terhadap serangan yang dilakukannya, mereka akan menggganti teks tadi dalam bentuk heksadesimal sehingga teks <script> akan berbentuk *%3C%73%63%72%69%70%74%3E*. Sidik untuk mengenali serangan XSS adalah:

```
/((\%3C)|<)((\%2F)|\/)*[a-z0-9\%]+((\%3E)|>)/ix
```

Karakter **((\%3C)|<)** dalam sidik diatas adalah karakter untuk mengenali kurung buka dalam bentuk ASCII atau dalam bentuk hexadecimal. Karakter **((\%2F)|\/)*** dalam sidik diatas adalah untuk mengenali karakter miring muka (*forward slash*) yang ada dipakai *closing tag* serta ekuivalen heksadesimalnya. Teks **[a-z0-9\%]+** - untuk memeriksa *alphanumeric string* didalam tag atau ekuivalen heksadesimalnya. Karakter **((\%3E)|>)** digunakan untuk mengecek untuk mengenali kurung tutup atau ekuivalen hexadesimalnya.

Selain dengan sidik diatas, serangan XSS dapat juga menggunakan teknik <img_src=> Untuk mengenali serangan XSS dengan teknik ini , dalam percobaan digunakan sidik sebagai berikut:

```
/((\%3C)|<)((\%69)|i|(\%49))((\%6D)|m|(\%4D))((\%67)|g|(\%47))[^\n]+((\%3E)|>)
```

Teks **(\%3C)|<)** digunakan untuk mengenali karakter kurung buka**.**Teks **(\%69)|i|(\%49))((\%6D)|m|(\%4D))((\%67)|g|(\%47)** digunakan untuk mengenali teks 'img' dalam aneka kombinasi ASCII. Teks **[^\n]+** untuk mengenali adanya sembarang karakter yang mengikuti <img.Teks **(\%3E)|>)** untuk mengenali karakter kurung tutup.

Dalam percobaan aturan Snort untuk mengenali serangan XSS adalah sebagai berikut:

```
============================================================
Rule Snort untuk serangan XSS
============================================================
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"NII Cross-site
scripting attempt"; flow:to_server,established; pcre:"/((\%3C)|<)((\%2F)|\/)*[a-
z0-9\%]+((\%3E)|>)/i"; classtype:Web-application-attack; sid:9000; rev:5;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"NII Cross-site
scripting attempt"; flow:to_server,established; pcre:"
```

```
/((\%3C)|<)((\%69)|i|(\%49))((\%6D)|m|(\%4D))((\%67)|g|(\%47))[^\n]+((\%3E)|>)";
classtype:Web-application-attack; sid:9000; rev:5;)
```

Simulasi serangan XSS ini dilakukan dengan car webclient memberikan permintaan sebagai berikut:

```
a.http:\\victim\nama.php?nama= <b>  'Test' </b>
b.http:\\victim\nama.php?nama=<script>alert('tes XSS')</script>
c.http:\\victim\nama.php?nama=<img_src>alert('tes XSS')</img_src>
```

Perintah :\\victim\nama.php?nama= <*b*> 'Test' </b> digunakan untuk menguji serangan XSS yang menggunakan teks HTML, sedangan perintah simulasi http:\\victim\nama.php?nama=<script>alert('tes XSS')</script> digunakan untuk serangan XSS yang menggunakan perintah script. Perintah http:\\victim\nama.php?nama=<img_src>alert('tes XSS')</img_src> digunakan untuk simulasi serangan dengan menggunakan teknik <img_src>.

Sistem pengirim informasi serangan telah berhasil mengirimkan pesan peringatan ke pesawat telepon genggam administrator dalam bentuk pesan , data hasil uji system dapat dilihat dalam Tabel 2.

Tabel 2. Hasil uji coba sistem terhadap simulasi serangan XSS

| Nama Serangan | Jam Serangan | Jam Kirim SMS | *Timeliness* | Operator Seluler |
|---|---|---|---|---|
| XSS attack 1 | 11.00 | 11.02 | 2 menit | Pro Xl |
| XSS attack 2 | 11.05 | 11.07 | 2 menit | Pro Xl |
| XSS attack 3 | 11.09 | 11.10 | 1 menit | Pro Xl |

Pada hasil pengujian ini, terbukti sistem telah berhasil dalam mengirimkan pesan peringatan ke Administrator saat sensor IDS mendeteksi adanya serangan. Rentang waktu yang terbentang antara saat terjadinya serangan dengan pengiriman SMS tentunya merupakan fungsi dari lalu lintas data operator seluler yang bersangkutan.

### 3.5. Pelanggaran kebijakan

Simulasi pelanggaran kebijakan dalam penelitian ini adalah adanya usaha untuk mengakses web site yang menyajikan informasi untuk pria/wanita dewasa. Dalam *mailing list bugtraq*, usaha akses situs dewasa ini dapat dikenal dengan adanya data yang mengandung kata-kata seperti *hardcore, fetish , young teen*, *tinygir*l, dan ungkapan yang sejenis. Guna mendeteksi usaha pengaksesan situs dewasa ini disusun aturan sebagai berikut:

```
#-------------------------------------
#ATURAN DETEKSI USAHA AKSES SITUS DEWASA
#-------------------------------------
alert tcp  169.254.223.99 $HTTP_PORT  -> HOME_NET ANY (msg:"TES USAHA SITUS PORNO 1";
flow:to_client,established; content:"HARDCORE"; nocase; classtype:kickass-porn; )
alert tcp  169.254.223.99 $HTTP_PORT  -> HOME_NET ANY (msg:"TES USAHA SITUS PORNO 2";
flow:to_client,established; content:"young teen"; nocase; classtype:kickass-porn; )
alert tcp  169.254.223.99 $HTTP_PORT  -> HOME_NET ANY (msg:"TES USAHA SITUS PORNO 3";
flow:to_client,established; content:"fetish"; nocase; classtype:kickass-porn; )
```

Tabel 3. Hasil uji coba sistem terhadap pelanggaran kebijakan

| Nama Pelanggaran | Jam Pelangaran | Jam Kirim SMS | *Timeliness* | Operator Seluler |
|---|---|---|---|---|
| Akses web dewasa 1 | 10.00 | 10.04 | 4 menitt | Pro Xl |
| Akses web dewasa 2 | 10.10 | 10.13 | 3 menit | Pro Xl |
| Akses web dewasa 3 | 10.14 | 10.16 | 2 menit | Pro Xl |

Hasil tanggapan sistem menunjukkan bahwa sistem berhasil menangkap kegiatan pelanggaran aturan ini dan melakukan pengiriman SMS ke Administrator. Tabel lengkap disaijkan dalam tabel Hasil Simulasi Pelanggaran Kebujakan dalam Tabel 3.

### 3.6. Pengujian modul respon

Pengujian modul respon bertujuan untuk mengetahui unjuk kerja modul respon. Modul respon merupakan modul yang bekerja berdasarkan perintah SMS dari administrator yang telah didefenisikan terlebih dahulu. Perintah tersebut tercantum dalam Tabel 4.

Tabel 4. Daftar perintah untuk modul respon

| No | Perintah SMS | Aksi |
|---|---|---|
| 1. | 0 | Mematikan server |
| 2. | 1 | Mereboot server |
| 3. | 2 | Mengunci server |
| 4. | 3 | Memutus hubungan jaringan |
| 5. | 4 | Menghubungkan ke jaringan internet |

Pada pengujian dilakukan simulasi serangan berupa usaha probing phpinfo(), dan admin melakukan perintah untuk mematikan server, mereboot, mengunci, memutus dan menghubungkan server ke jaringan. Hasil pengujian ditampilkan dalam Tabel 5.

Tabel 5. Hasil uji modul respon

| No. | Jenis simulasi serangan | Waktu Simulasi | Jenis Perintah | Waktu pengiriman perintah | Waktu eksekusi perintah |
|---|---|---|---|---|---|
| 1. | Probing PHP | 10.00 WIB | Mematikan Server | 10.05 WIB | 10.13 WIB |
| 2. | Probing PHP | 10.30 WIB | Reboot | 10.38 WIB | 10.43 WIB |
| 3. | Probing PHP | 11.00 WIB | Mengunci server | 11.05 WIB | 11.12 WIB |
| 4. | Probing PHP | 11.25 WIB | Memutus koneksi | 11.30 WIB | 11.34 WIB |
| 5. | Probing PHP | 12.00 WIB | Menghubungkan | 12.12 WIB | 12.18 WIB |

Hasil pada Tabel 5 menunjukkan bahwa modul respon telah bekerja dengan baik.

## 4. SIMPULAN

Berdasarkan data hasil pengujian simulasi serangan yang dilakukan menunjukkan nilai *timeliness* rata-rata adalah sebesar 1,67 menit, dan lebih baik dari yang dihasilkan Fawcet [3] yang berkisar pada angka 13 jam. Hasil pengujian modul respon, menunjukkan bahwa penggunaan notifikasi SMS menaikkan unjuk kerja dari sudut *timeliness* dan memberikan kemampuan admin melakukan tindakan yang perlu dari jarak jauh, yang tidak dapat diperoleh dari sistem notifikasi dengan file log atapun email. Pada penelitian ini telah dapat dilakukan penggabungan sistem notifikasi SMS ke dalam sistem IDS, sehingga akan meningkatkan keterawasan jaringan dan meningkatkan keandalan jaringan.

## DAFTAR PUSTAKA

[1]. Yegeneswaran, ”**Internet: Global Characteristic and Prevalence**”, Proceeding of the 2003 ACM Sigmetrics International Conference on Meus, 2003.
[2]. CERT Statistic, htttp:/www.cert.org, 2002.
[3]. Fawcet, T. and Foster, P., ”**Activity Monitoring: Noticing Interesting Change in Behaviour**”, Proceeding of the fifith ACM SIGDD International Conference of on KDD, New York ACM Press, San Diego, 1999.
[4]. Baruffi, R., Michela, M., and Montanari, “**Planning for Security Management**”, IEEE Security System and Their Application, Los Alamitos, CA, 2002.
[5]. Dobrucki, M., “**Prioritie in the Development of Network Instrusion Detetction System**”, Master Thesis, Helsinki University of Technology, 2002.
[6]. Godal, J, et.al., ”**The Work of the Instrusion Detection Rethinking The Role of Security Analisis**” ,Proceedings of the Tenth Americas Conferences on Information Systems, New York, 2004.
[7]. Kristopher K., "**A Database of Computer Attack for The Evaluation of Intrusion Intrusion Detection System**", Master Thesis, MIT, 1999.
[8]. Kumar, J.D., "**Attack Developmnet for Intrusion Detection**", Master Thesis, MIT, 2000.
[9]. Sandeep, K., "**Classification and Detection of Computer Intrusion**", Doctoral Thesis, Purdue University, 1995.